

这个星期完成了许多事情，包括：

1、完善了ResourceType。新增

RESOURCE_TYPE_1DDB_GRAPHICS_TEXTURE , RESOURCE_TYPE_2DDB_GRAPHICS_TEXTURE , RESOURCE_TYPE_3DDB_GRAPHICS_TEXTURE , RESOURCE_TYPE_1DGEN_GRAPHICS_RENDER_TARGET , RESOURCE_TYPE_2DGEN_GRAPHICS_RENDER_TARGET , RESOURCE_TYPE_3DGEN_GRAPHICS_RENDER_TARGET, 分别是OpenGL的Texture和RenderTarget。

2、实现了OpenGL的Texture3D, 并且实现了该资源的网络传输。

3、实现了并行光线投射体绘制，不包含合成。

这是截止星期二的工作，然后实现了RenderTarget2D资源。并且实现了该资源的网络传输。

接着新建了一个Operation命名为OperationCompose, 用于并行结果的合成。可是因为涉及到多机之间的数据传输，发现了许多问题：

- 1、在RenderTarget被绘制完成后，过早地被释放了，没有来得及合成操作。于是调试的时候很奇怪地发现所有资源都不见了，开始以为是调试器有问题，后来跟踪才发现是那些资源被释放了，消耗了不少时间。这个问题可以通过增加释放资源的延时来解决。
- 2、网络传输RenderTarget和绘制RenderTarget使用了不同的线程，在OpenGL中必须通过切换current opengl context来解决。但是这样会增加并行计算模型的复杂度。所以需要对RenderTarget进行重新设计，让数据常驻于内存而不是显存，这样就可以用网络传输了。具体是绘制完后，马上将数据传输到内存，然后释放显存空间，以备网络传输之用。
- 3、如果1中的解决方案不起效，那么需要一套错误响应的机制。通知Master进程，某个Unit的执行出错。

下面是星期二的效果图，体数据是自动生成的球体。合成效果需要到下个星期才能做。



